Securing DevOps: Where to start and what to measure 🔐👩🏽‍💻📈

Stefania Chaplin @devstefops

Solutions Architect @ Gitlab

Credit to Image Creators

Agenda

- #whoami
- What?
- Who?
- How?
- Why?
- Summary
- Q&A

Quiz

#whoami 🦄

👩🏽‍💻

Python, Java
Rest APIS

🔐

DevSecOps
AppSec, CloudSec

GitLab

The DevOps
Platform

🏄🏽‍♀️ ☯️ 🌴

What?

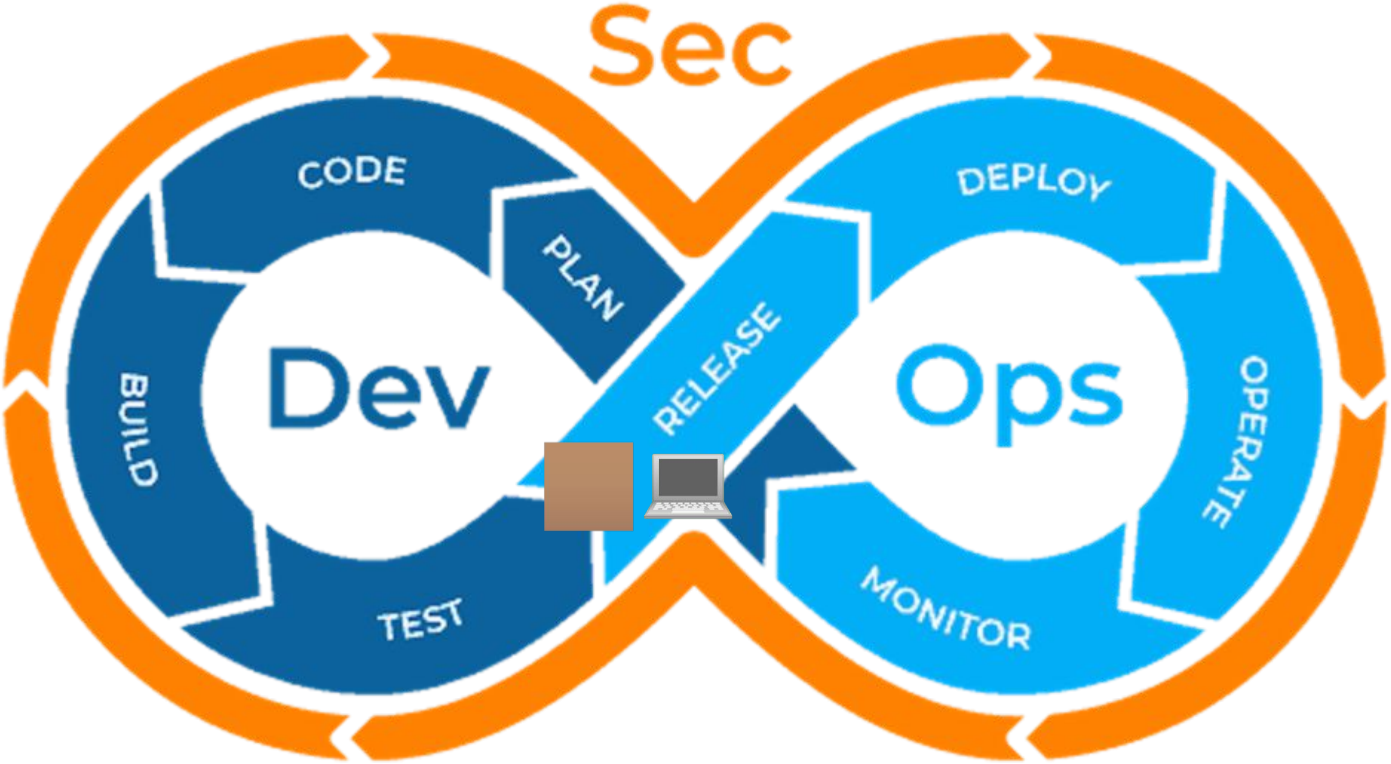
Quiz

What Is DevSecOps?

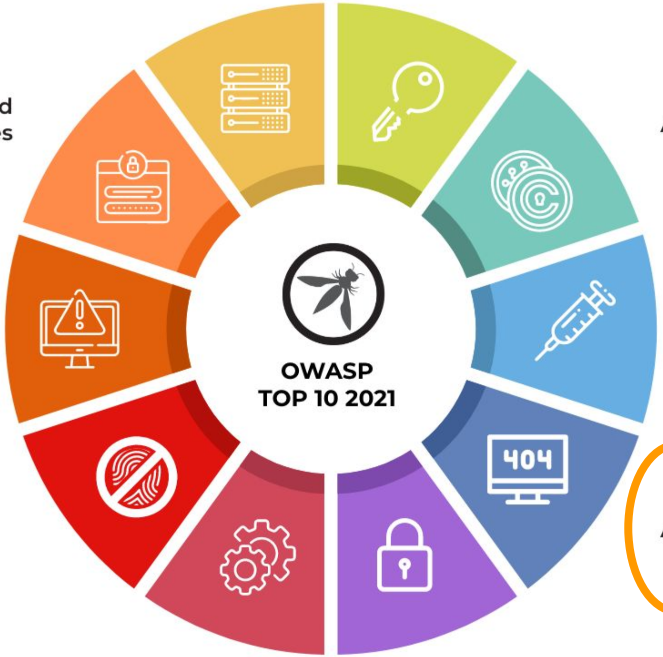
DevSecOps stands for **dev**elopment, **sec**urity, and **op**eration**s**.

👩🏽‍💻 🔐 🔧

It's an approach to **culture, automation**, and **platform design** that **integrates security** as a **shared responsibility** throughout the **entire IT lifecycle.**


Sec
CODE
PLAN
BUILD
Dev
TEST
RELEASE
Ops
DEPLOY
OPERATE
MONITOR

Web



A01. Broken Access Control

A02. Cryptographic Failures

A03. Injection

A04. Insecure Design

A05. Security Misconfiguration

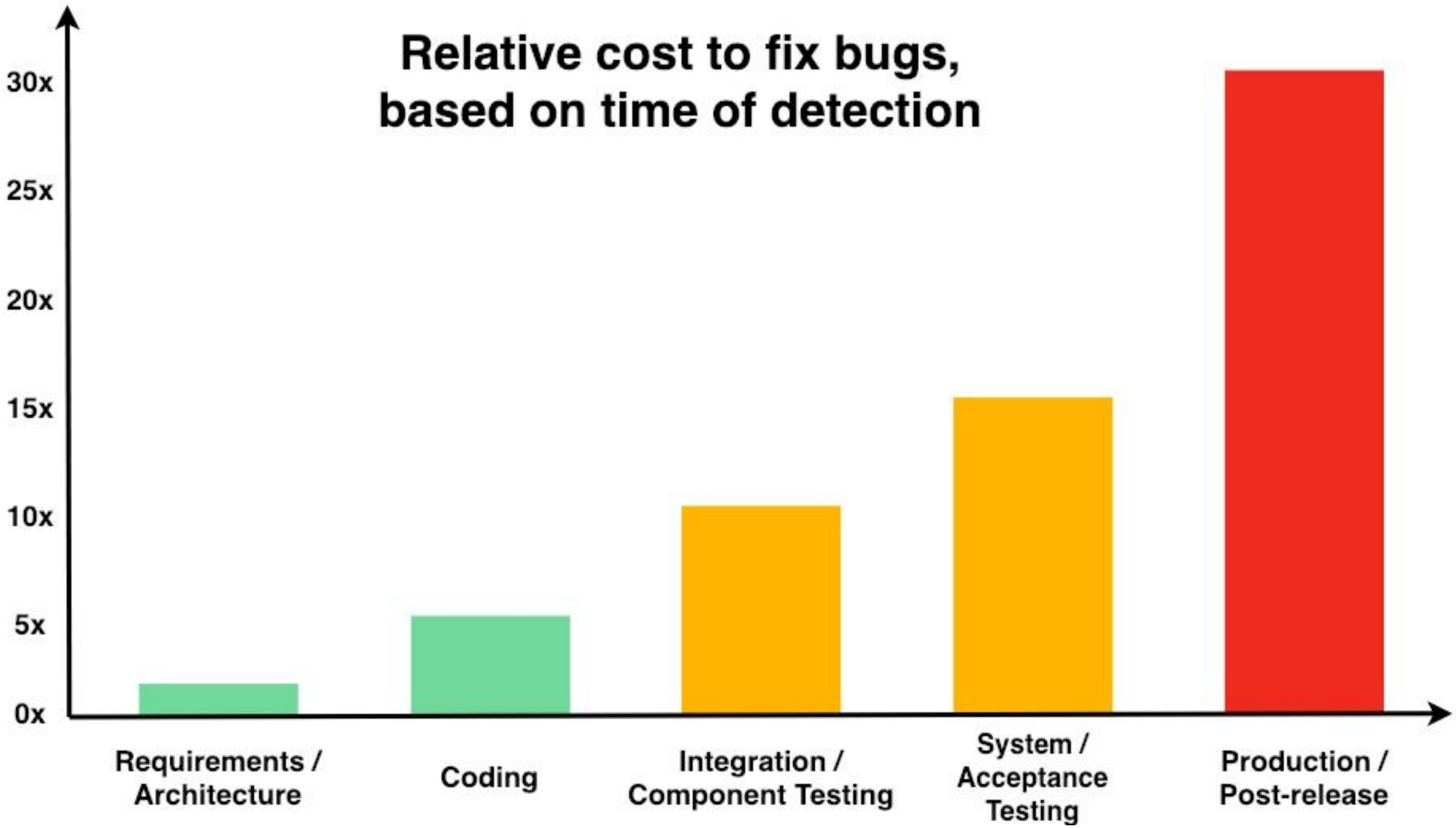
A06. Vulnerable and Outdated Configurations

A07. Identification and Authentication Failures

A08. Software and Data Integrity Failures

A09. Security Logging and Monitoring Failures

A10. Server - Side Request Forgery



Relative cost to fix bugs, based on time of detection

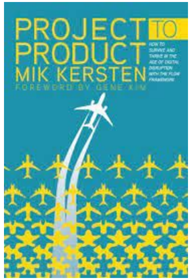
Common Pain Points 💥

- Security is the bad guy 👻
- Vulnerabilities (known + unknown) make it to production 🚨
- Delays, fails, or…. ‘worse’ 📉

Who?

Quiz

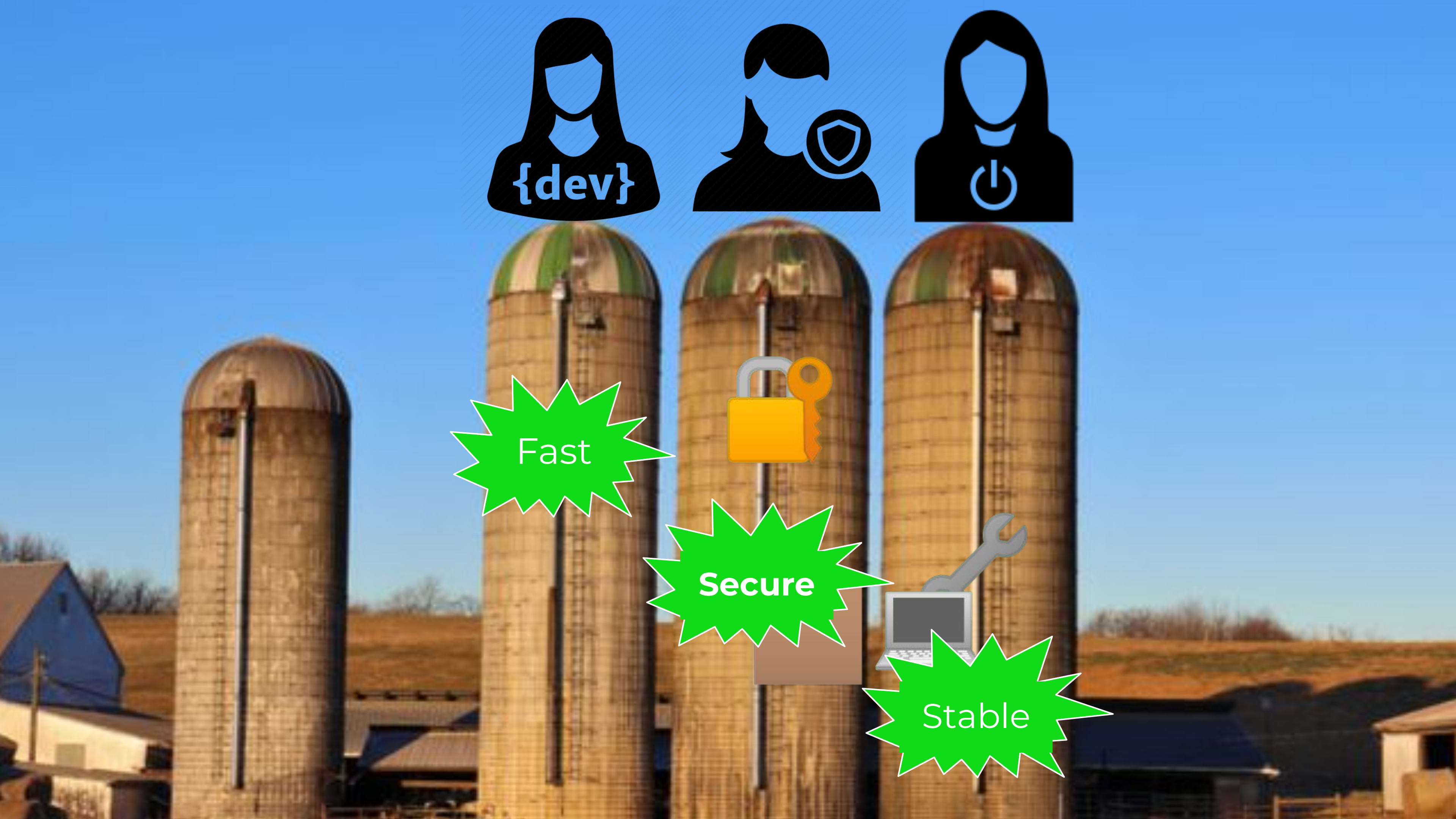
What is the difference between project mindset vs product mindset ?



PROJECT TO PRODUCT

MIK KERSTEN

FOREWORD BY GENE KIM

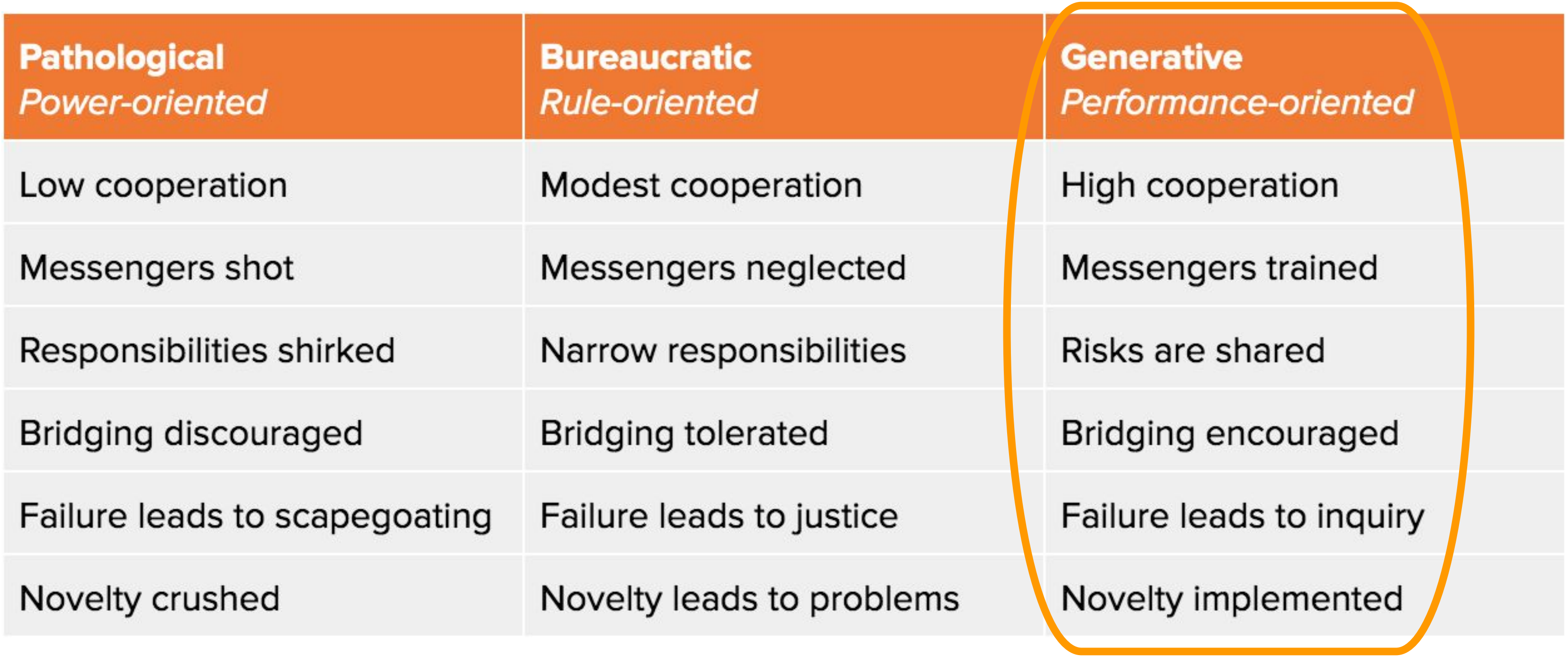


Fast

Secure

Stable

What is Culture? ☯



Pathological *Power-oriented*	**Bureaucratic** *Rule-oriented*	**Generative** *Performance-oriented*
Low cooperation	Modest cooperation	High cooperation
Messengers shot	Messengers neglected	Messengers trained
Responsibilities shirked	Narrow responsibilities	Risks are shared
Bridging discouraged	Bridging tolerated	Bridging encouraged
Failure leads to scapegoating	Failure leads to justice	Failure leads to inquiry
Novelty crushed	Novelty leads to problems	Novelty implemented

Warstrum 2004

{dev}
HARVESTORE
UNSTABLE

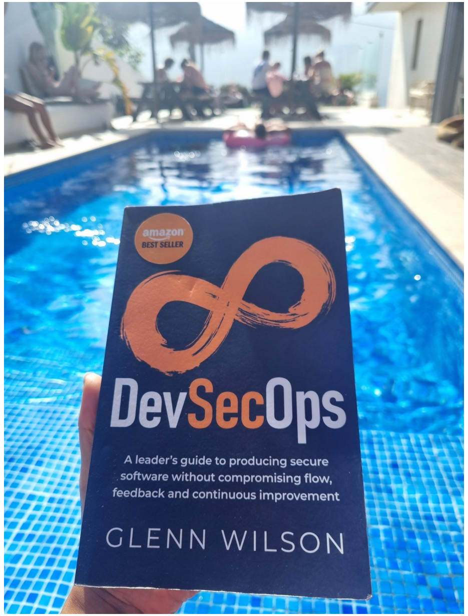
How?

Quiz

Where do we start Securing DevOps?

Some useful reading 📚

- 🧠 Education
- 🎨 Secure by Design
- 🚀 Automation

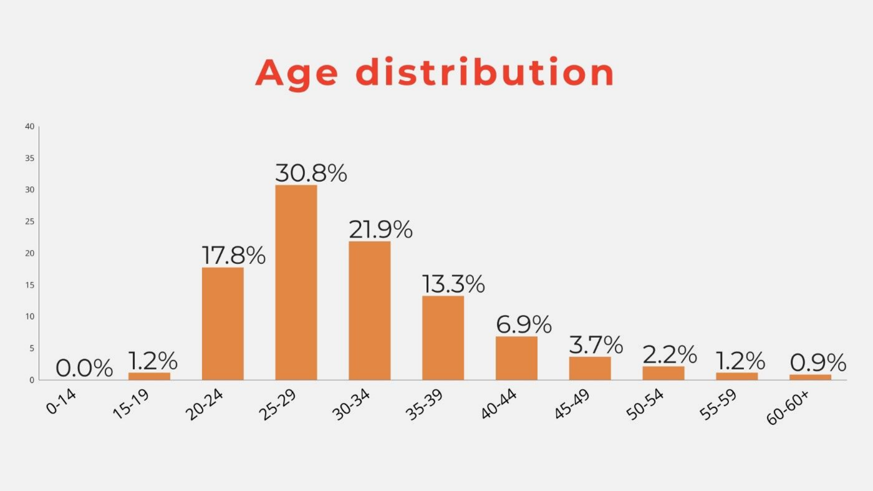


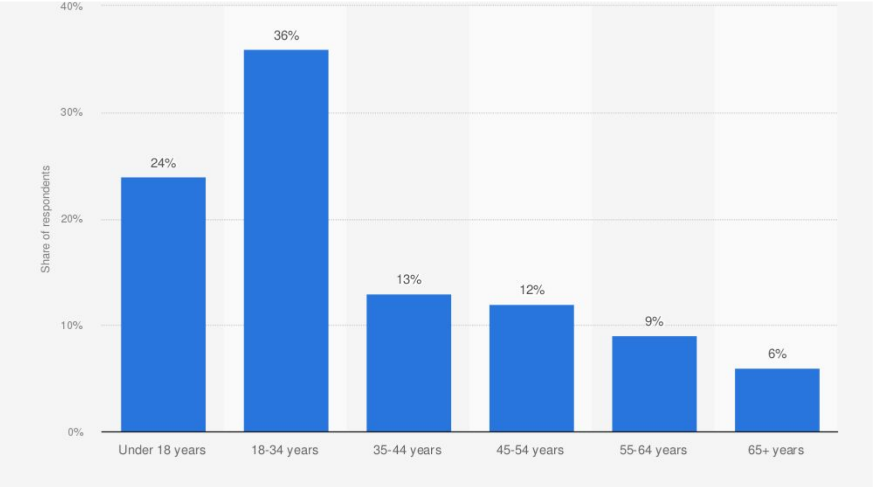
DevSecOps

A leader's guide to producing secure software without compromising flow, feedback and continuous improvement

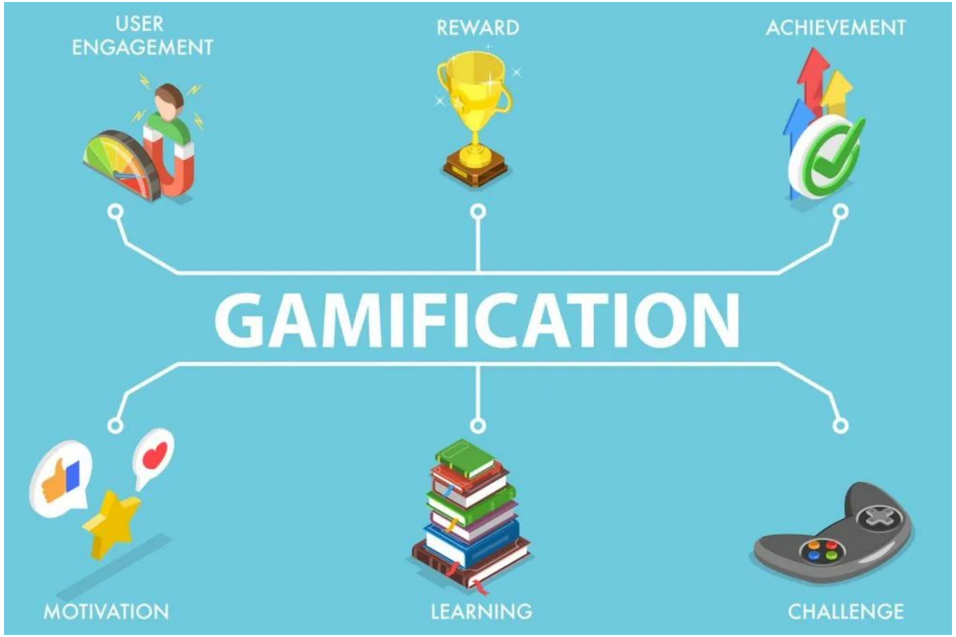
GLENN WILSON

Developers or Video Gamers? 🤓🎮

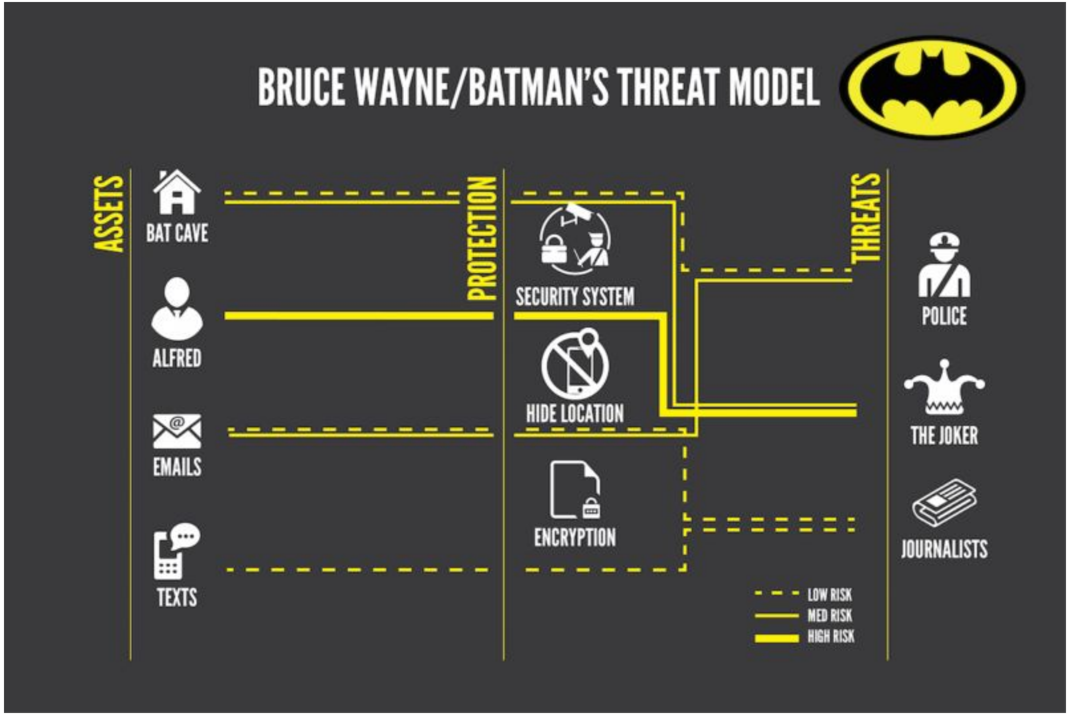




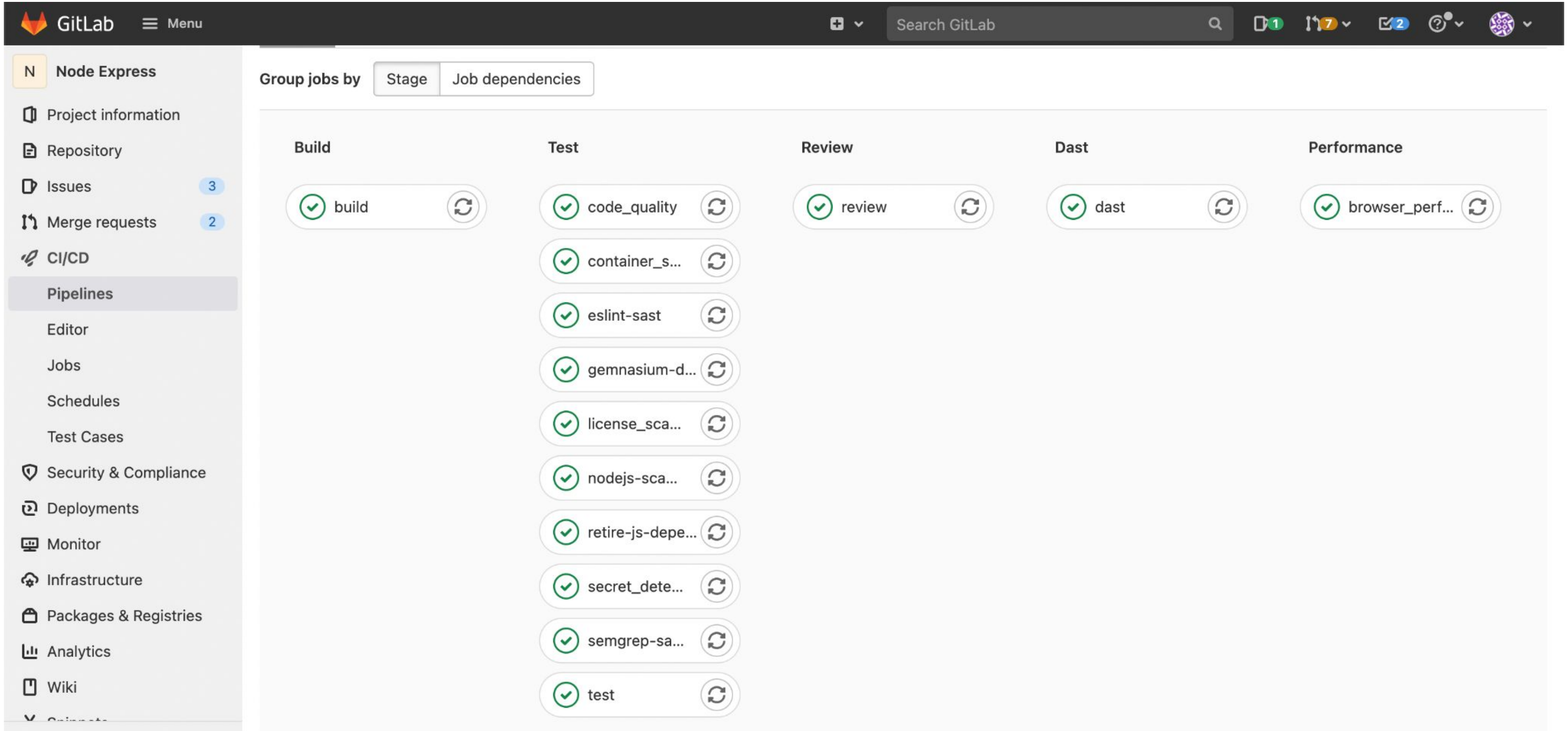
Make Security Fun & Easy 😄



Make Security Fun & Easy 🦇



Shift Security Left 🔐👉



Group jobs by Stage | Job dependencies

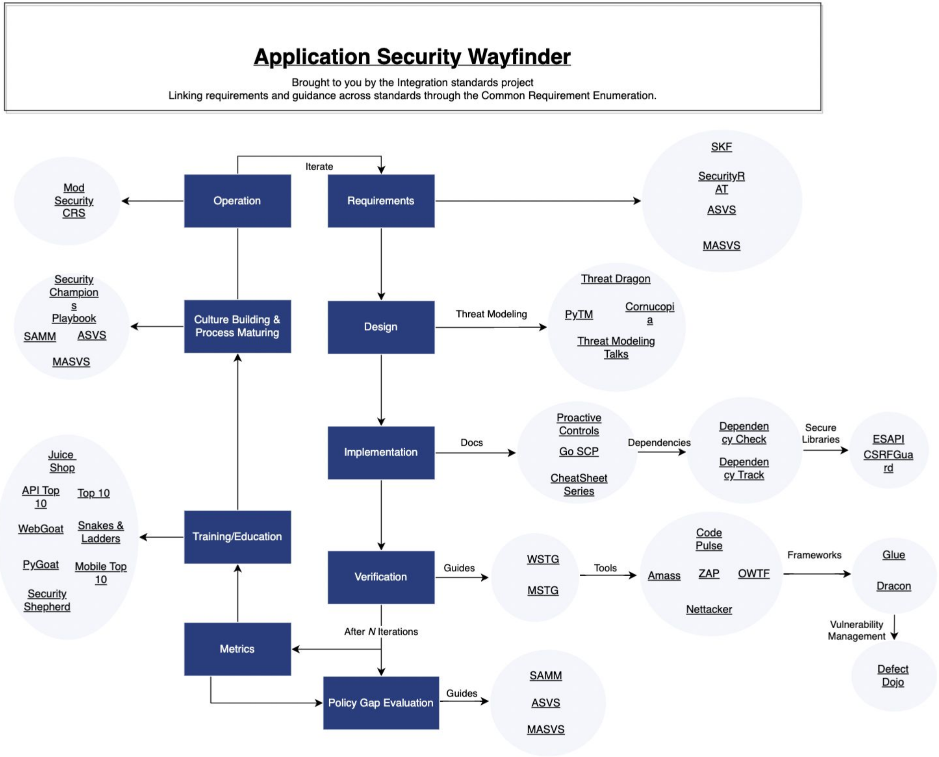
Build	Test	Review	Dast	Performance
✓ build	✓ code_quality	✓ review	✓ dast	✓ browser_perf...
	✓ container_s...			
	✓ eslint-sast			
	✓ gemnasium-d...			
	✓ license_sca...			
	✓ nodejs-sca...			
	✓ retire-js-depe...			
	✓ secret_dete...			
	✓ semgrep-sa...			
	✓ test			

Your Friendly Neighbourhood OWASP

OWASP
Zed Attack Proxy

CycloneDX

OWASP
Dependency-Check



Application Security Wayfinder

Brought to you by the Integration standards project
Linking requirements and guidance across standards through the Common Requirement Enumeration.

Operation → Mod Security CRS

Iterate: Operation → Requirements

Requirements → SKF, SecurityRAT, ASVS, MASVS

Requirements → Design

Culture Building & Process Maturing → Security Champions Playbook, SAMM, ASVS, MASVS

Design → Threat Modeling → Threat Dragon, PyTM, Cornucopia, Threat Modeling Talks

Design → Implementation

Implementation → Docs → Proactive Controls, Go SCP, CheatSheet Series → Dependencies → Dependency Check, Dependency Track → Secure Libraries → ESAPI, CSRFGuard

Training/Education → Juice Shop, API Top 10, Top 10, WebGoat, Snakes & Ladders, PyGoat, Mobile Top 10, Security Shepherd

Implementation → Verification

Verification → Guides → WSTG, MSTG → Tools → Code Pulse, Amass, ZAP, OWTF, Nettacker → Frameworks → Glue, Dracon → Vulnerability Management → Defect Dojo

After *N* Iterations: Verification → Metrics, Policy Gap Evaluation

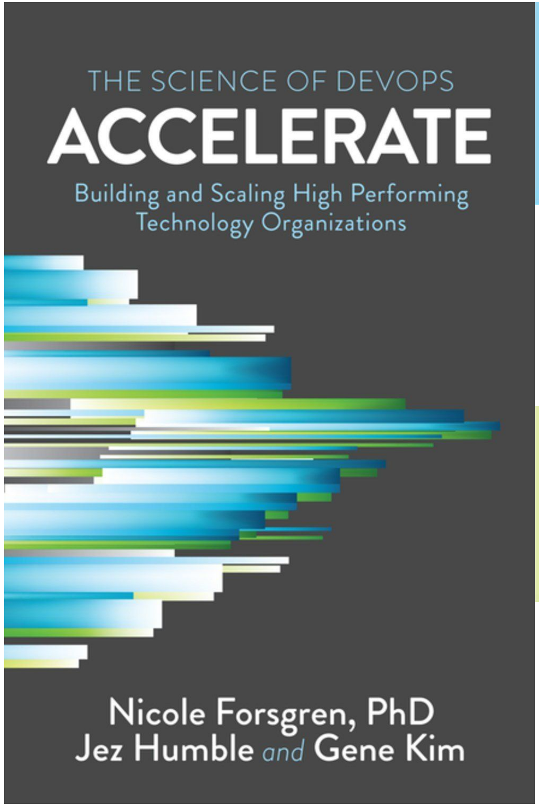
Metrics → Training/Education → Culture Building & Process Maturing

Metrics → Policy Gap Evaluation → Guides → SAMM, ASVS, MASVS

Quiz

How do we measure DevSecOps?

DORA Metrics 🚀 ⚖️

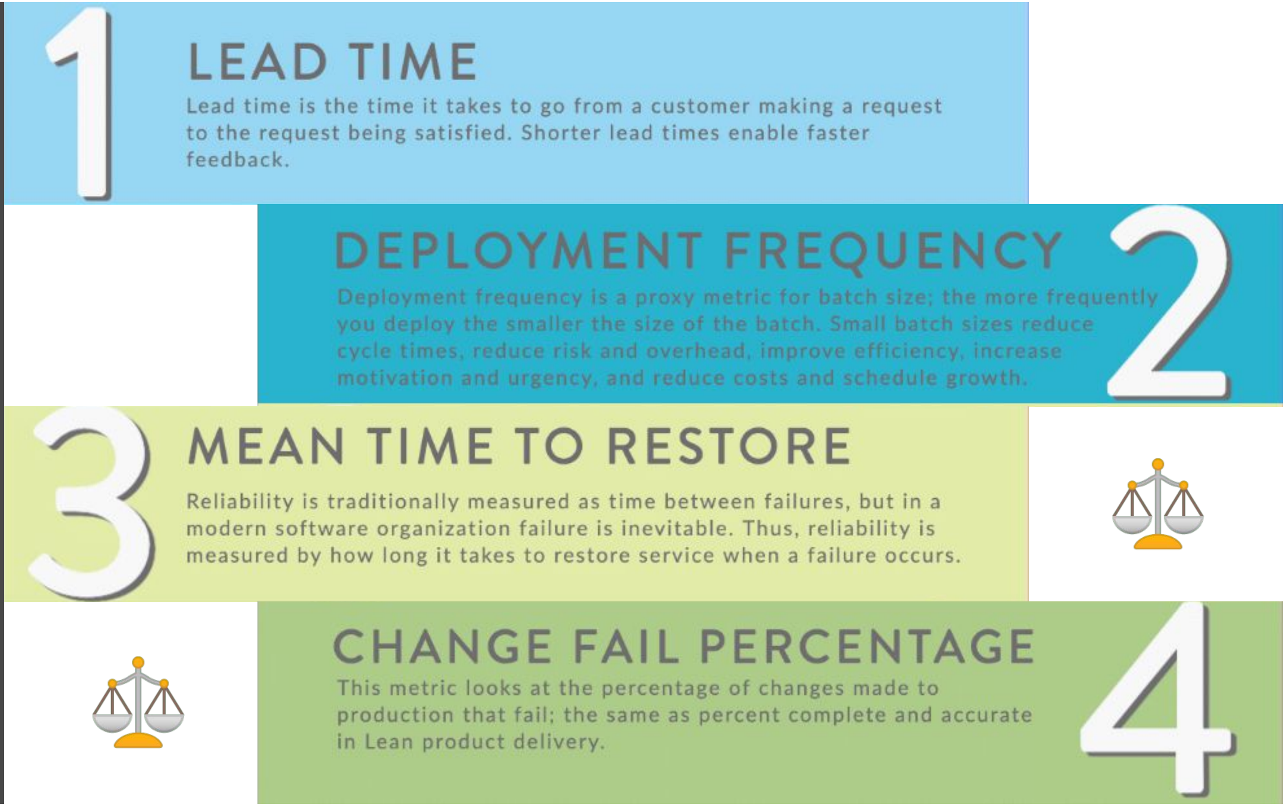


THE SCIENCE OF DEVOPS

ACCELERATE

Building and Scaling High Performing Technology Organizations

Nicole Forsgren, PhD
Jez Humble *and* Gene Kim



1 LEAD TIME

Lead time is the time it takes to go from a customer making a request to the request being satisfied. Shorter lead times enable faster feedback.

2 DEPLOYMENT FREQUENCY

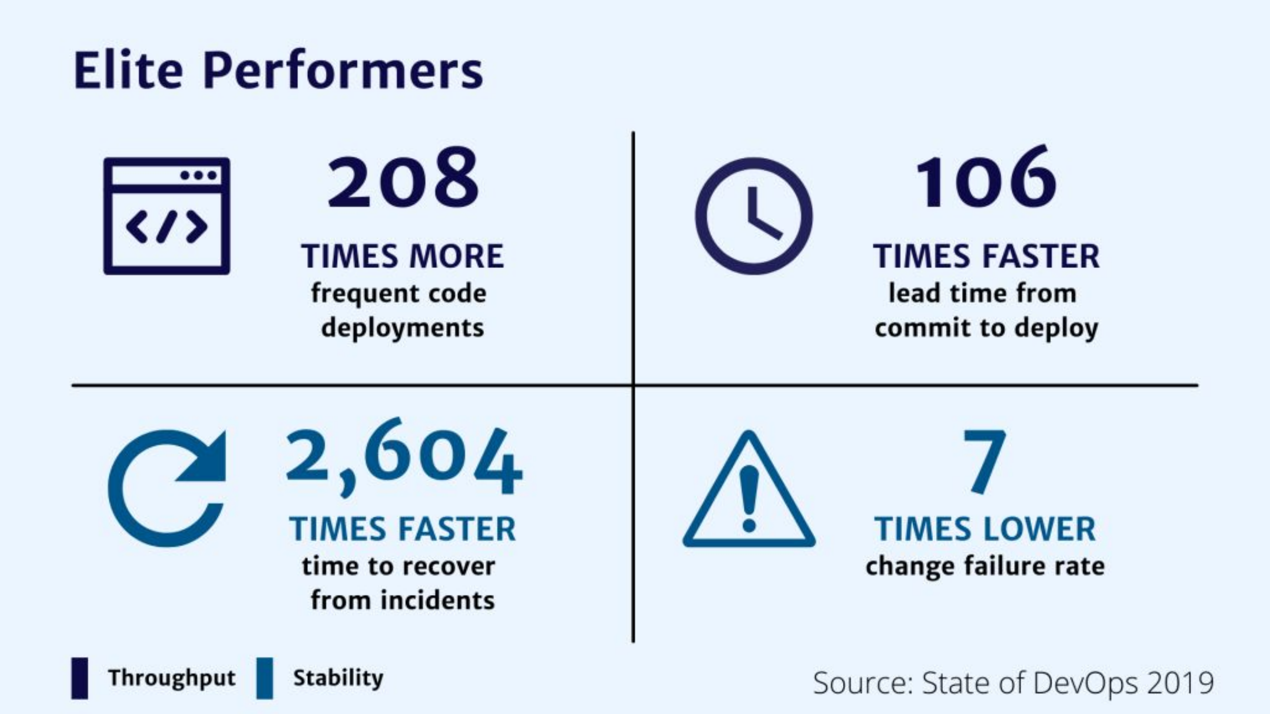
Deployment frequency is a proxy metric for batch size; the more frequently you deploy the smaller the size of the batch. Small batch sizes reduce cycle times, reduce risk and overhead, improve efficiency, increase motivation and urgency, and reduce costs and schedule growth.

3 MEAN TIME TO RESTORE ⚖️

Reliability is traditionally measured as time between failures, but in a modern software organization failure is inevitable. Thus, reliability is measured by how long it takes to restore service when a failure occurs.

4 CHANGE FAIL PERCENTAGE ⚖️

This metric looks at the percentage of changes made to production that fail; the same as percent complete and accurate in Lean product delivery.



Elite Performers

208
TIMES MORE
frequent code deployments

106
TIMES FASTER
lead time from commit to deploy

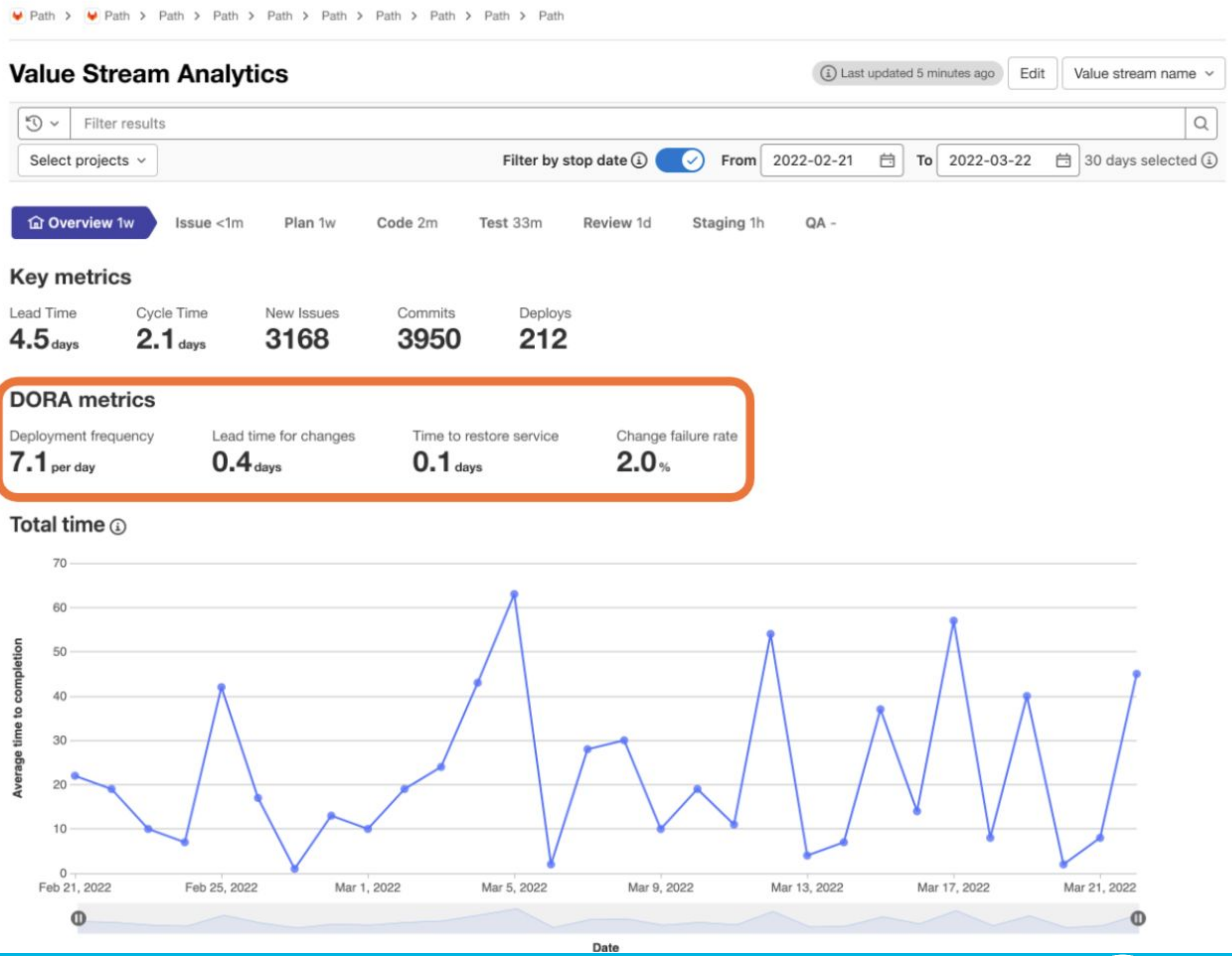
2,604
TIMES FASTER
time to recover from incidents

7
TIMES LOWER
change failure rate

▮ Throughput ▮ Stability

Source: State of DevOps 2019

‘Elite performers spend **50% less time** remediating security issues than low performers’



Path > Path > Path > Path > Path > Path > Path > Path > Path > Path

Value Stream Analytics

Last updated 5 minutes ago | Edit | Value stream name

Filter results

Select projects | Filter by stop date ☑ | From 2022-02-21 | To 2022-03-22 | 30 days selected

Overview 1w | Issue <1m | Plan 1w | Code 2m | Test 33m | Review 1d | Staging 1h | QA -

Key metrics

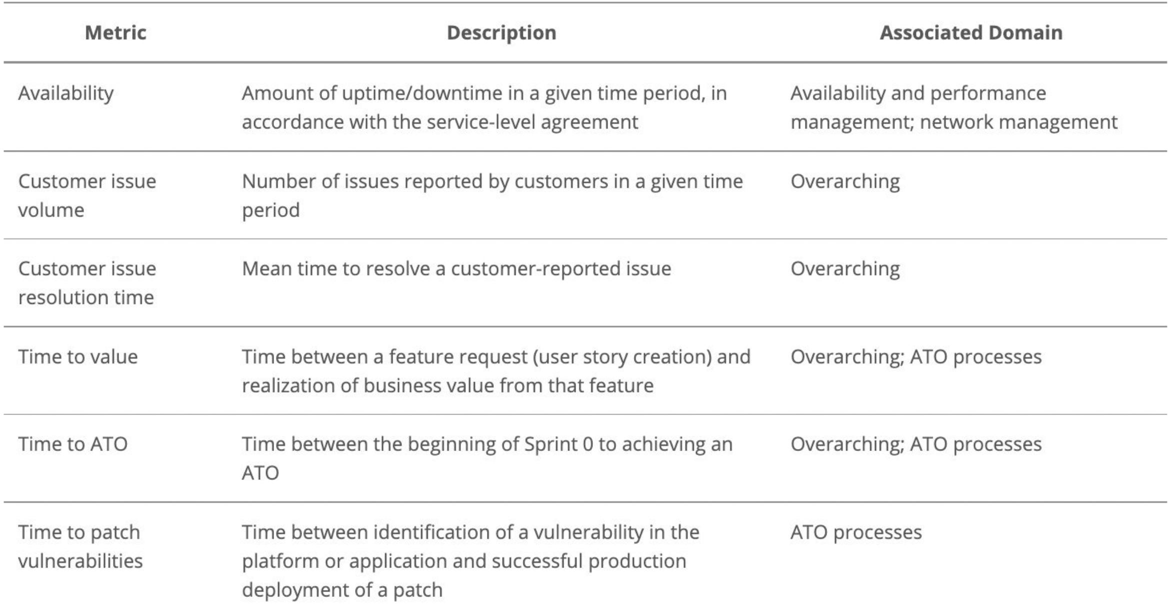
Lead Time	Cycle Time	New Issues	Commits	Deploys
4.5 days	2.1 days	3168	3950	212

DORA metrics

Deployment frequency	Lead time for changes	Time to restore service	Change failure rate
7.1 per day	0.4 days	0.1 days	2.0 %

Total time

Beyond Dora Metrics 📏



Metric	Description	Associated Domain
Availability	Amount of uptime/downtime in a given time period, in accordance with the service-level agreement	Availability and performance management; network management
➡️ Customer issue volume	Number of issues reported by customers in a given time period	Overarching
➡️ Customer issue resolution time	Mean time to resolve a customer-reported issue	Overarching
Time to value	Time between a feature request (user story creation) and realization of business value from that feature	Overarching; ATO processes
Time to ATO	Time between the beginning of Sprint 0 to achieving an ATO	Overarching; ATO processes
➡️ Time to patch vulnerabilities	Time between identification of a vulnerability in the platform or application and successful production deployment of a patch	ATO processes

https://insights.sei.cmu.edu/blog/the-current-state-of-devsecops-metrics/

Why?

Why DevSecOps? 🔐

- Incorporating security into DevOps helps speed up iterations, we can innovate faster than competitors 🚀
- Vulnerabilities are identified earlier which helps to avoid cyber-attacks 💥
- It helps improve communication and collaboration between teams 🤝

https://developer.ibm.com/articles/devsecops-what-and-why/

Summary

- Take a #securityfirst approach 🧠
- Break down silos, we are all on the same team! 🌐
- Make it fun, automate & measure results #empowerdevelopers 🦸🏽‍♀️

Thank you!

stefania@devstefops.com